

Shubham Sharma

Senior Software Quality Engineering (QE) Professional

QE Specialist with 5+ years of experience in engineering quality, reliability, and observability into monoliths and microservices. Brings hands-on experience across both corporate and startup environments, contributing to projects of diverse scales, architectures and user bases. Proven expertise in a wide range of testing disciplines, including **UI, API, accessibility, visual, performance, database** and **mobile** testing. Skilled in test automation, planning, documentation, defect management, and performance optimisation utilising tools like **Playwright, Puppeteer, Cypress, Selenium WebDriver, k6, Pact, Postman, Appium, Jest, React Testing Library, JUnit, Percy, Lighthouse, LambdaTest, BrowserStack, Sauce Labs, HotJar, Grafana, Jira, Confluence, X-ray, Zephyr, Slack**. Skilled in applying context-driven, risk-based, and first-principles thinking to build scalable, efficient, and adaptive testing strategies, test suites and tooling.

Demonstrates a deep understanding of **DevOps**, fostering seamless collaboration via pair and mob programming sessions with developers to streamline **SDLCs**. Possesses strong software engineering experience, proficient in a broad range of technologies including **JavaScript, TypeScript, HTML, CSS**, scripting languages like **Bash, Python**, and VSC/laC tools such as **Git, GitHub, Jenkins**. Experienced with cloud platforms **AWS, Azure, GCP**, laC tools like **Terraform, Ansible**, and containerisation technologies such as **Docker, Kubernetes** to automate deployments, enhance scalability, and improve troubleshooting capabilities.

Highly effective communicator with a proven track record of identifying, debugging, and resolving complex issues through strong collaboration, stakeholder management and root cause analysis. Recognised for exceptional problem-solving and defect tracking abilities, consistently delivering cost-effective solutions by leveraging automation and “as-code” paradigms. Committed to continuous learning and adapting to emerging technologies and evolving tech stacks, thriving in both collaborative team settings and independent work environments.

Australian Citizen | Available in 2 Weeks | +61452558430 | Sydney, NSW, 2000 | shubhamsharma.emails@gmail.com | [Testimonials](#) | [Portfolio](#) | [LinkedIn](#) | [GitHub](#) | [Linktree](#) | [Medium](#) | [Substack](#) | [Academic Portfolio](#) | [NPM Packages](#)

WORK EXPERIENCE

Senior Software Quality Engineer Qantas

08/2023 - Present | Mascot, NSW, Australia

Own and lead the **Software Quality Engineering (SQE)** domain of Team Concierge, overseeing the development of the Qantas and Jetstar Hotels UIs and APIs. Define and execute long-term quality and testing strategies that align with business objectives, customer expectations, industry best standards, and compliance requirements, ensuring high-impact deliverables meet the highest standards. Utilised **Postman** for comprehensive API testing, ensuring seamless integration between the UI and backend services.

Architect and continuously optimise automation test suites, particularly unit (**Jest**) and end-to-end (**Cypress** and **Playwright**) test suites, with a focus on context-driven coverage, leveraging libraries that enhance efficiency and quality (such as **jest-axe**, **allure-playwright** and **cypress-parallel**), and ensuring test suites are lean and maintainable to accelerate development cycles. Leverage understanding of **Java** to use **WireMock** to setup a mock server for UI and API test automation.

Champion quality throughout SDLCs, empowering and guiding the Concierge team on testing activities, technological best practices and fostering cross-functional collaboration. Break down silos and cultivate a culture of shared ownership of product quality, innovation and operational excellence, while identifying and eliminating bottlenecks, inefficiencies, and risks. Collaborate closely with leadership to drive continuous improvement to the software, product, business, processes and delivery. Actively contributed to discussions on API design and management, leveraging **Zero Trust Architecture** to ensure robust and secure APIs.

Lead initiatives to enhance the team's **DORA** metrics, resolve technical debt, drive data-informed decisions, and conduct root cause analysis. Oversee infrastructural health checks, ensure observability of dashboarding artifacts, and establish performance benchmarks across projects. Manage efforts from pre-discovery stages, such as product planning, through scaling, including cross-team initiatives.

Accomplishments:

- Centred focus on unrealised dimensions of quality such as compatibility, SEO metrics and performance, and identified

gaps utilising **Lighthouse** reports of critical webpages

- Introduced new automated testing methodologies - Snapshot Testing with **cypress-image-diff-js** and **Smart UI (LambdaTest)** and Accessibility Testing with **jest-axe**, **cypress-axe**, **axe-playwright**, **pa11y-dashboard** and **pa11y-ci**, finding a total of 160 accessibility violations across both platforms and 8 main web pages (each).
- Ensured full automated accessibility test coverage was met by shifting left on automated accessibility testing and introducing **axe-core/react** to display warnings in the console during development, **jest-axe** to validate the accessibility of **React** components, **eslint-plugin-jsx-a11y** to check for accessibility violations during and **pa11y-ci** to check for a11y violations on SIT to prevent the accessibility violations from reaching production.
- Established complete **AA Conformance to Web Content Accessibility Guidelines 2.2** by organising continuous sessions with users to conduct rigorous testing using assistive technologies like screen readers (**NVDA, JAWS**), voice recognition software (**Dragon NaturallySpeaking**), and specialised input devices (such as trackballs) across 10 main web pages. Incorporated user feedback to resolve over 200 accessibility issues, improving the website's overall accessibility score from 60% to 98%. Received positive testimonials from disabled users praising the enhanced user experience from follow-up sessions. Implemented best practices, such as keyboard navigation optimisation and dynamic content accessibility, setting new accessibility standards for Qantas Group.
- Directed test coverage improvements across all testing dimensions, such as for Unit Testing, [by ensuring 100% of Unit Test coverage was being met across the production codebase](#), and developed [Bash script](#) to fail CI builds if the Unit Test coverage dropped due to a deployed change.
- Orchestrated the creation of a comprehensive acceptance criteria for unit testing, encompassing code coverage thresholds, test case design guidelines, test execution, test organisation and structure, maintainability best practices, reporting and documentation requirements, performance and efficiency considerations and test environment management, CI practices and industry and regulatory compliance.

- Championed a drive to overhaul the existing unit test suite, rectifying 64 flaky unit tests (out of 908) through meticulous review and automated detection mechanisms via [@smartesting/flaky-test-detector](#).
- Delivered drastic improvements to CI Test Feedback Loop for the Unit Test Suite (**Jest**) by using shards and parallel workers, and E2E Test Suite (**Cypress**) using **cypress-parallel** and removing duplicate logic across tests, resulting in [Jest tests in the CI pipeline running 6 times faster](#) and [Cypress tests in the CI pipeline running 2 times faster](#).
- Acted as the voice of the customer, identified and regularly implemented UI/UX improvements to polish and enhance the customer experience, such as making footer text ('Terms and Conditions') collapsible in mobile view to reduce clutter.
- Regularly led and scheduled iterative and collaborative Spike meetings to drive process, testing and CI/CD improvements, such as requiring Devs to use a Spell Checking IDE extension (**Code Spell Checker**), to prevent all future spelling mistakes from reaching production and updating [PR templates](#) to include mandatory testing section.
- Led initiative to clean up all spelling mistakes across both Hotels UI repositories using codespell, [fixing a total of 58 spelling mistakes](#), and integrated **cspell** into **ESLint** to block CI branches containing spelling mistakes from merging.
- Introduced additional Code Quality Checkers in CI: **knip**, **type-check**, **prettier**, **pa11y**, **PMD**, **fossa** and **webhint**.
- Spearheaded partnership with Product Design to address quality considerations earlier and improve Design-Development processes, such as building a **Slack** parity bot to automatically notify Product Designer when a change was ready for design review and introducing the use of screenshot comparison tool **ashot** in development workflows to compare code changes against **Figma** specifications and reduce padding margin misalignments, incorrect colour selections, etc.
- Spearheaded the merging of Qantas and Jetstar Hotels UI repositories into single monorepo, leveraging **Turbo** and **pnpm**, to reduce overhead of needing to clone PRs, context switch, and provide a unified source of truth.
- Implemented continuous testing by creating a comprehensive set of synthetic tests in **Datadog** to perform User Experience (UX) monitoring and browser-based health checks for critical and main web pages across both Hotels platforms. These tests were configured to run every ten minutes, ensuring that the response time is under 8000 milliseconds, the status code matches 200 (OK), and the header 'content-type' is set to 'text/html; charset=utf-8'. Alerts were established to notify a dedicated Slack channel whenever any test failed, facilitating prompt identification and resolution of issues.
- Led the adoption of **semantic-release** to establish a systematic and automated release process, transforming our continuous deployment workflow from ad-hoc releases to consistent and high-quality application updates. By leveraging **semantic-release**, a standardised versioning scheme was implemented, release creation and publishing were automated, and comprehensive release notes were generated on an automated basis. This resulted in a more efficient and reliable release pipeline, which increased team productivity and allowed us to deliver value to our users more frequently and predictably.
- Led the adoption of **Playwright** over **Cypress**; configured Playwright to run E2E tests for both platforms in the monorepo, and migrated **Cypress** tests into the monorepo with **cypress-to-playwright** and manual modifications.
- Established more robust quality gates at critical milestones like completion of core booking functionality, integration testing, and release candidates in order to prevent defects from flowing downstream.
- Formalised a checklist-based release certification processes during Product Verification Tests and Post-implementation Reviews including validation of critical integrations, defining acceptance criteria for UAT aligned to user stories, establishing key performance metrics, and mandating accessibility, security, and compliance testing gates before production releases.
- Conducted UAT sessions before major releases, driving requirements alignment and identifying gaps in user stories.

- Brought user feedback from **Hotjar** into design reviews, raising flags on, and streamlining complex user flows.
- Led the implementation of **Sentry**'s 'User Feedback' widget feature to resolve unidentified and uncaught bugs.
- Reduced bug resolution time by 40% through strategic configuration of **Sentry** with sourcemap integration

Senior Software Test Analyst 3P Learning

06/2022 – 08/2023 | North Sydney, NSW, Australia

Oversaw and directed the spearheading of manual and automated Software Quality Assurance (SQA) initiatives and implementations for Team Yeti, responsible for the development of the UI designed for teachers and parents of 3P learning's SaaS e-learning product lineup. Performed Static Test reviews of user stories, business requirements and functional specifications. Collaborated with developers to define and implement mechanisms to inject testing earlier into the SDLC, such as by implementing a systematic process of validating UI changes before merging to staging using a locally hosted web server. Leveraged **Cypress Studio** to automatically generate **Cypress** test scripts alongside manual test execution, streamlining test development by reducing boilerplate code, accelerating test extension, and enabling stakeholders to create tests without relying on traditional **BDD** tooling like **Cucumber**.

Performed deployments with, and configured automated tests to execute reliably and efficiently in **Buildkite**. The **Buildkite** pipeline leveraged **Kubernetes** for dynamic scaling of test runners, ensuring efficient utilisation of resources and faster feedback loops. Contributed to the configuration and maintenance of the **Kubernetes** deployments, ensuring the stability and scalability of the testing infrastructure. Led test pipeline triage rotations in **Buildkite** and ensured pipeline failures were promptly debugged and resolved. Enabled rapid feedback loops by prioritising parallel execution of **Buildkite** jobs across **Buildkite** agents, leveraging **Buildkite**'s Concurrency Gates. Routinely executed post-release, manual and automated regression testing on production, sent out release notes emails and updated the company-wide release notes changelog. Alerted developers and stakeholders when discovering issues and bugs on **Slack**, along with detailed **Jira** tickets containing screenshots of **Chrome**; **Firefox**; **Safari**—**DevTools** console; network error logs, and/or HAR files, and/or URLs of Rollbar items.

Accomplishments:

- Reported multiple security issues and malpractices, such as the absence of an in-built timeout session function for the MFA page of the customer SSO login portal page.
- Repeatedly reported back multiple ambiguities and gaps in User Stories, Functional Specifications and Acceptance Criteria to the Product Owner, resulting in improved requirements documentation, reduced development rework, accelerated product delivery, and heightened user satisfaction.
- Engineered an end-to-end **Cypress** automation suite that revolutionised the manual release pipeline, automating build identification and selection from **Buildkite** while streamlining deployment workflows across staging and production environments. Implemented automated release notes capture and documentation, successfully eliminating manual intervention throughout the release process and significantly reducing deployment overhead.
- Refactored 15 out of 45 existing **Cypress** Tests, speeding up the E2E Test Suite feedback loop by an average of 24%.

Software Engineer eesel AI

12/2022 - 06/2023 | Sydney, NSW, Australia

Collaborated directly with co-founders and go-to-market team to rapidly prototype AI-powered solutions and marketing tools in a high-velocity startup environment. Delivered full-stack web applications and AI integrations using **TypeScript**, **Python**, and emerging AI technologies to accelerate product-led growth initiatives.

Accomplishments:

- Accelerated content production 5x by architecting an AI-powered workflow with **Python** that automatically generated SEO-optimised blog content, significantly reducing manual content creation overhead.
- Modernised digital presence by rebuilding eesel's marketing website with **Next.js**, **TypeScript**, **GraphQL** and **Vercel**, implementing scalable architecture and **WordPress CMS** integration that improved site performance and reduced iteration cycles.
- Drove product adoption by developing a Chrome extension that integrates with eesel's AI platform to generate intelligent email and chat responses, directly supporting go-to-market strategy and increasing user engagement with core product features.

Junior Software Test Analyst Australian Community Media (ACM)

09/2021 - 06/2022 | Sydney CBD, NSW, Australia

Co-managed the priorities and direction of the QA function of the Project Engineering team. Implemented Software Quality Engineering Best Practices into developer and product operations and performed manual and automated UI testing of ACM's websites and web application systems to ensure the highest quality UI/UX standards were met. Created POCs for the tooling and automation solutions. Built and used Playwright test framework and CI/CD pipeline that used **Docker**, **Terraform**, and **AWS**. Collected artifacts such as Defect Trends (**Jira**), Error Logs (**Datadog**), User Feedback (**Hotjar**) and more, to help prioritise and advocate for SQA initiatives. Followed **BDD** principles, such as directing the team towards the adoption of the GWT structure when writing test cases and user stories. Spearheaded the adoption of the [Honeycomb Test Model](#) to instrument testing and enhanced the granularity and precision of testing efforts. Provided input and estimates to the development of business systems solutions and specifications. Spearheaded quality strategy discussions, identified gaps, and highlighted potential risks in the product roadmap.

Accomplishments:

- Assisted the onboarding of new SQA engineers onto the team.
- Implemented a roadmap and structure for automated; feature, sanity and smoke testing processes.
- Advocated and implemented measures for wider-scaled testability and observability of web application systems; such as configuring **Grafana** dashboards for early warning of traffic spikes on high-profile articles.
- Contributed to evolving testing practices, such as leading the adoption of the honeycomb test model over the pyramid test model, the Jira-integrated test management tool **Zephyr** over Google Sheets and **Playwright** over **Cypress**.
- Guided the development of a Quality Mindset within the team, through meetings and documentation, leading to a surge of non-QE members bringing up ideas and solutions of their own to drive quality-oriented improvements.
- Supported training and development initiatives, enabling focus on gaps in team processes and habits.
- Regularly created meetings to coordinate the raising and resolving of incidents and issues to facilitate the continuous deployments of error-free production releases.

Software Tester GoGet CarShare

04/2021 - 09/2021 | Artarmon, NSW, Australia

Performed manual and automated, UI and database testing of GoGet's web and mobile application. The types of functional testing performed included regression, smoke, sanity, integration, end-to-end and UAT. The types of non-functional testing performed included stress, reliability, portability, compatibility, stability and usability. Operated and maintained a UI test automation framework leveraging **Selenium WebDriver**, **TestNG**, and **Maven**. Oversaw manual execution along with continuous execution of the test suite, and continuously maintained **TestNG** artifacts like **TestNG** test classes, **TestNG.xml** and **pom.xml**.

Managed CI/CD pipeline with **Bamboo**, and used **Docker** for containerisation to enable **consistent** testing and deployments across low-level environments. The Bamboo pipeline was integrated with **Azure DevOps** to leverage its robust reporting and tracking capabilities, providing enhanced visibility into the testing process and facilitating better collaboration among team members. Leveraged **Azure's** cloud infrastructure for scalability and cost-effectiveness in running the automated test suites. Authored and executed **SQL** queries within **MySQL** server to check for the data population in application tables to verify data and perform testing against **Oracle** database. Authored and maintained test cases, data, reports and plans. Documented and reviewed user stories with stakeholders, and ensured test execution was in line with test plans. Assisted the project manager with sprint planning and defect management efforts, and clarified ambiguous requirements to ensure the effectiveness of test plans.

Accomplishments:

- Identified, documented, reported and resolved over 40 bugs, interoperability flaws and UI/UX-related issues.
- Validated, communicated, and resolved application-related issues through thorough documentation and testing to maximise the return on investment of web application development initiatives
- Contributed to successful production releases through comprehensive testing that helped to maintain GoGet CarShare's globally industry-leading sales ranking, market share and customer-satisfaction rating.
- Always employed a "test-to-break" approach on testing initiatives.
- Improved the documentation of functional test suites to speed up test execution, enhance the repeatability of test cases, and bolster testing standardisation.
- Routinely held meetings to collaborate and build strong rapport with developers and project stakeholders.

Test Administrator Applause

07/2020 - 49/2021 | Sydney CBD, NSW, Australia

Led crowdsourced product testing initiatives for Meta's Ray-Ban Stories Smart Glasses, orchestrating voice command data collection to optimise voice control functionality. Managed testing operations in a high-velocity environment to support development of this innovative wearable technology.

Led and supported diverse product testing teams, managing end-to-end testing coordination including: orchestrating testing sessions across multiple user groups. Documenting comprehensive test outcomes and analyzing results. Developing and maintaining an internal database of tester profiles and qualifications. Ensuring smooth testing operations through effective scheduling and communication.

Executed comprehensive prototype testing for Ray-Ban Stories Smart Glasses, utilising terminal operations and **Android Debug Bridge (ADB)** to conduct rigorous dry-run testing and troubleshooting protocols. Extract and process voice command recordings through **Z shell**. Perform manual data analysis and validation. Manage secure data transfers to Meta's servers.

Accomplishments:

- Consistently surpassed operational KPI targets while managing and supporting 10+ product testers daily, maintaining exceptional customer satisfaction ratings
- Innovated testing efficiency by developing and implementing a custom [Bash script](#) that streamlined troubleshooting and data extraction workflows for Ray-Ban Stories Smart Glasses prototypes. Achieved 32% improvement in testing efficiency within 3 weeks of deployment. Earned formal recognition from Meta leadership for initiative and impact. Reduced manual processing time and improved data accuracy through automation.

EDUCATION

Bachelor's Degree in Computer Science | Western Sydney University

EXPERTISE

Development Methodologies: Scrum, Kanban, Extreme Programming | Waterfall, Spiral, Iterative, V-Model

Development Techniques: TDD, BDD, Pair Programming, Mob Programming

Programming Principles: KISS, YAGNI, DRY, SINE, AHA, OCF, SRP, SoC

Programming Paradigms: OOP, FP, Procedural Programming, Reactive Programming, Literate Programming, Functional Reactive Programming

Mental Models: Lean Thinking, First-Principles Thinking, Second-Order Thinking, Systems Thinking, Lateral Thinking, Probabilistic Thinking, Design Thinking

Software Architecture/Design: Singleton, Factory, Observer, MVC, Component-Based Architecture, Hexagonal Architecture, Clean Architecture

Testing/Quality Assurance: UI, API, Mobile, Database, Visual, Performance, Security, Accessibility Testing | Decision Table Testing, Data Driven Testing, Keyword Driven Testing, Modular Driven Testing, Page Object Modelling, Equivalence Partitioning, Boundary Value Analysis, State Transition Testing, Use Case Testing

Branching Strategies: Gitflow, Feature Branching, Trunk-Based Development, Release Branching

Compliance/Standards: ISO 25000, GDPR, PCI-DSS, WCAG 2.2

TECHNICAL SUMMARY

Unit Testing: Jest, ViTest, Mocha, Chai, JUnit, NUnit, PyTest, Jasmine, RSpec, Shoulda, minitest, test-unit, Enzyme, React Testing Library

UI Testing: TestNG, Cypress, Playwright, Selenium WebDriver, WebdriverIO, TestCafe, Puppeteer, Robot Framework, CodeceptJS, Wallaby.js, Selenium, Cucumber, Capybara, Watir, HtmlUnit, Ghost Inspector, Telerik Test Studio, SnapTest, Eggplant Test, Mabl, Testim.io, Gauge, Galen Framework, PhantomJS, Katalon Studio

Mobile Testing: Appium, Detox, Espresso, XCUITest, Maestro, WebdriverIO, EarlGrey

API Testing: Postman, RestAssured, SuperTest, Karate, Karate DSL, Bruno, Insomnia REST, ReadyAPI, Axios, ky, Dredd, Tavern, Apickli, Schemathesis, GraphQL Test, Apollo GraphQL, Newman, Chakram

Contract Testing: Pact, PactFlow, Specmatic, Dredd

Visual Regression Testing: Percy, Applitools, Chromatic, BackstopJS, Pixelmatch

Accessibility Testing: axe-core, jest-axe, axe-playwright, Pa11y, WAVE, Lighthouse, NVDA, JAWS

Performance/Load/Stress Testing: JMeter, k6, Artillery, Gatling, Locust, BlazeMeter, LoadRunner, NeoLoad, NBomber, SoapUI, xk6-browser, Tsung

Security Testing: OWASP ZAP, Burp Suite, Wireshark, Fiddler, Metasploit, SQLMap, Arachni, w3af, Vega, cURL, HTTPie, Nmap

Mocking/Service Virtualisation: MSW, json-server, Sinon.js, Mockito, WireMock, Mountebank, MockServer, Hoverfly

Infrastructure/Environment Testing: Terratest, InSpec, Serverspec, Testinfra

Database Testing: Testcontainers, H2 Database, DbUnit, Flyway, Liquibase

Cross-Browser Testing: BrowserStack, Sauce Labs, LambdaTest

Test Reporting/Analytics: Allure, ExtentReports, ReportPortal, TestRail, Zephyr, Xray, Qase, qTest, PractiTest, TestLink, SpiraTest, QMetry, TestFort, Squash, Plutora

Code Quality/Static Analysis: ESLint, Prettier, Stylelint, CodeQL, SonarQube, CodeClimate, Checkmarx, Veracode, Dependabot, type-check, knip

Test Data Management: Faker.js, Factory Boy, Mockaroo

Programming Languages: JavaScript, TypeScript, HTML, CSS, SCSS, SASS, JSON, YAML, Bash, PowerShell, Python, Java, Kotlin, Groovy, Scala, Golang, C, C++, C#, Ruby, PHP, Haskell, SQL, Rust, Swift, Elixir, Clojure, Erlang, REST, GraphQL, gRPC

Frontend Frameworks/Libraries: React, Vue.js, Angular, Svelte, Next.js, Nuxt.js, Gatsby, Remix, SolidJS, Lit, Alpine.js, jQuery, Ember.js

Backend Frameworks: Node.js, Express.js, NestJS, Deno, FastAPI, Flask, Django, Spring Boot, Ruby on Rails, Laravel, Symfony, ASP.NET Core, Gin, Fiber

CSS Frameworks/Styling: Tailwind CSS, Bootstrap, Material-UI, Ant Design, Chakra UI, Bulma, Foundation, Radix UI, shadcn/ui, React Aria, Styled Components, Emotion, Normalize.css, PostCSS

State Management: Redux, Redux Toolkit, Zustand, MobX, Vuex, Pinia, NgRx, Recoil, Jotai, Context API

Build Tools/Bundlers: Webpack, Vite, Rollup, Parcel, esbuild, Turbopack, SWC, Babel, Turbo, pnpm, npm, Yarn

Databases/Data Storage: PostgreSQL, MariaDB, SQLite, Oracle, SQL Server, MongoDB, Redis, Elasticsearch, Cassandra, CouchDB, DynamoDB, InfluxDB, Neo4j, NoSQL, MS Access, Citrix, SQLMap

Version Control/Collaboration: Git, GitHub, GitLab, BitBucket, SVN

Configuration Management: Dotenv, Ansible, Chef, Puppet

Developer Tools/Utilities: Lodash, Zod, Nodemon, UML, TestFlight, Electron, Storybook, Swagger, OpenAPI

IDEs/Environments: Visual Studio, Android Studio, Eclipse, WezTerm, Vim, Neovim, Emacs, Bash, Z shell, PowerShell, WSL

CMS/CDN: Contentful, WordPress, Sanity.io, Adobe Experience Manager, Contentstack, Google Cloud CDN, Cloudflare, Akamai

Containerisation/Orchestration: Docker, Kubernetes, Terraform, Vagrant, VMware, VirtualBox

CI/CD: Buildkite, GitHub Actions, GitLab, Jenkins, BitBucket

A/B Testing/Analytics: Split.io, Optimizely, GA4, HotJar

Cloud Computing: AWS, Azure, GCP, Cloudflare, Vercel

Defect Management: Jira, Shortcut, Linear, ClickUp, Trello, Asana

Observability/APM: Datadog, Sentry, Splunk, Rollbar, Grafana